Dense Approximate Storage in Phase-Change Memory

Jacob Nelson, Adrian Sampson, and Luis Ceze
University of Washington
{nelson, asampson, luisceze}@cs.washington.edu

ABSTRACT

Multi-level phase-change memory stores bits by quantizing the resistance value of each cell. For example, a PCM cell with four distinct resistance levels can store two bits. By dramatically increasing this granularity, high density can be achieved at the cost of storage reliability. Many applications—those that deal with sensory data or are otherwise resilient to errors—can take advantage of this increased density with only small impact on overall quality of service. We present a very basic evaluation that suggests that approximate PCM storage may be a good option for lossy image storage.

1. INTRODUCTION

Emerging resistive storage technologies such as phase-change memory (PCM) store digital data by quantizing an analog signal. That is, a PCM cell stores data by varying the electrical resistance of a material. A single-bit cell might store a 1 using a high resistance value and a 0 using low resistance. Using more than two distinct resistance levels, a PCM cell can store more data in a single cell, resulting in a higher-density storage medium.

Typically, PCM's granularity (and thus its information density) is limited by the precision with which a particular resistance can be set and read in the cell. For example, if we could precisely set 256 distinct resistance levels in a PCM cell, we could store 8 bits per cell. However, two adjacent resistance values in the continuum of 256 values are likely to be so close that noise in the material's resistance properties are likely to prohibit reliable reading and writing in the cell.

PCM designs usually view this noise as a *limit* on density. However, it is also possible to see the situation as a *trade-off* between density and reliability. Applications that do not require perfect storage fidelity for some of their data may exploit this trade-off for significant gains in storage density.

A significant body of recent work (for example, [1, 4, 5]) has observed that many "soft computing" applications can tolerate occasional errors with minimal detriment to output quality. For example, graphics applications that deal with lossily compressed images implicitly incorporate imperfect data storage. In general, applications that deal with sensory data (e.g., images, audio, video) or inherently imprecise values (e.g., in machine learning algorithms) can tolerate imperfect data storage. However, even these applications can typically only tolerate errors in *some* of their data. An image file, for instance, consists of a small amount of critical, error-sensitive header and metadata information along with a large amount of error-tolerant pixel data. So hardware that supports approximation should be configurable to support both approximate and precise operation.

We propose PCM devices that permit both low-density, high-reliability storage and high-density, low-reliability storage. Such devices could significantly reduce the storage footprint of soft computing applications while having negligible impact on their quality of service.

2. APPROXIMATE STORAGE IN PHASE-CHANGE MEMORY CELLS

We take advantage of a configurable multi-level cell (MLC) PCM array as in [6]. MLC devices can typically trade off read and write latency for density.

An MLC PCM array supporting approximation provides more resistance levels than are reliably writable or readable. That is, approximate PCM cells are vulnerable to two types of errors: *write errors*, in which the cell sets the resistance of the material incorrectly, and *read errors*, in which the cell is not able to accurately measure the material's resistance within the time allotted for a read operation.

2.1 Writing to an approximate MLC

Data is stored in a PCM cell by controlling the crystallization of the cell's chalcogenide through the application of a shaped write pulse to the cell's heating element. The same pulse may lead to a different resistances when applied to different cells in the array, or even to the same cell at different times [2].

A common technique to mitigate this variation is to add a verification step to the write process, making it iterative [3]. After a write pulse completes, the value of the cell is read and compared with the value being written. If the values are not the same, new write pulse parameters are computed and the cell is erased and rewritten. This process continues until the value read matches the value to be stored.

While this technique enables precise storage of values, the number of write iterations required to store a precise value increases with the number of bits stored in the cell. Thus, precise MLC leads to increased write latency, energy usage, and heater wear.

An approximate array can trace precision for write cycles. If an error bound ϵ is provided along with the value to be written, iteration can stop once the value stored is within ϵ

JPEG compression

approximate PCM storage

Figure 1: Two versions of the same image, one compressed using JPEG and the other stored in our simulated approximate PCM device. The two files occupy approximately the same number of PCM cells, where each cell is either single-level and precise (for JPEG) or multi-level and approximate (PCM).

of the value to be stored. For some applications, an error bound may not be necessary, and writes could be allowed to complete after a single iteration.

2.2 Reading from an approximate MLC

Data is read from a PCM cell by biasing the cell and using an analog-to-digital converter (ADC) to sense its resistance. A variety of ADC techniques might be used in constructing a MLC PCM array [6]. An integrating ADC counts the time required to discharge a capacitor through the cell. Its read time scales exponentially with the number of bits stored in the cell. A successive approximation ADC uses a sample-and-hold circuit to refine its measurement over multiple cycles. Its read time scales linearly with the number of bits stored in the cell.

An approximate array can trade precision for read latency. Again, an error bound can be provided with the read operation, and using the parameters of the ADC, this bound can be converted to a time limit. The ADC can be run until this time limit is up, after which the ADC's approximation of the value stored in the cell can be returned.

Note that different error bound may be used in different reads to the same cell. For instance, an image editing application applying a filter to an image might do low-latency approximate reads to generate a preview, and high-latency precise reads to generate the final result.

3. EVALUATION

We built a simple simulator to explore image storage in approximate MLC PCM. We modeled three types of errors. For write errors, we added a random integer taken from a normal distribution to the value stored in each cell. For read errors, we truncated bits read from each cell. We modeled errors due to heater wear by replacing the value in randomly-chosen subset of cells with a random value.

Figure 1 compares an image compressed using the standard JPEG compression technique with an uncompressed image stored approximately with our simulator. We stored 8 bits per PCM cell, allowing a write error standard deviation of 3, truncating one bit on reads, with wear errors in 1 out of 10,000 cells. The image occupies 768K cells. The JPEG-compressed image was compressed using the default quality level, and would occupy 659K cells in a single-level PCM device.

While we see few differences between the images, we also applied `perceptualdiff`, a common perceptual metric, to evaluate the difference more robustly [7, 8]. When compared against the original uncompressed image, the JPEG-compressed image shows 48,409 significant pixel differences, while the approximate image shows only 45,637.

4. CONCLUSION

Phase-change memory presents an opportunity for exploitation of a density-quality trade-off in data storage for soft computing applications. Our initial exploration suggests that unreliable PCM may be useful for lossy image storage.

At its core, the idea of unreliable storage addresses the fact that applications frequently do not require all the reliability that devices currently guarantee. In situations where perfect quality of service is unnecessary or even impossible, systems are effectively overprovisioned with reliability. By trading off perfect reliability for various benefits, many systems may perform better. Some work has already explored this design space in CPUs, where processors that expose some transient faults (such as timing faults) to software can show significant power and performance gains [1, 4].

Similarly, networks could benefit from the same principle: for transmission of some data (such as voice or video traffic), networks' error-correction schemes may be unnecessary. Especially in wireless networks, unnecessary correction of these errors can incur large power and performance costs. With access to the application-level error tolerance of individual packets, wireless networks could selectively avoid costly error correction or retransmission.

The potential for principled approximation in storage devices, CPUs, and networks represents an opportunity for general exploration into selectively reliable computer systems. These systems should permit both precise and approximate operation and expose this decision to the application programmer. Approximation-aware applications could then exhibit gains in storage density, performance, power, and other constraints beyond what is possible when correctness guarantees are strict.

5. REFERENCES

[1] M. de Kruijf, S. Nomura, and K. Sankaralingam. Relax: an architectural framework for software recovery of hardware faults. In *ISCA*, 2010.

[2] M. K. Q. et al. Improving read performance of phase change memories via write cancellation and write pausing. In *HPCA*, 2010.

[3] T. N. et al. Write strategies for 2 and 4-bit multi-level phase-change memory. In *IEDM*, 2007.

[4] A. Kahng, S. Kang, R. Kumar, and J. Sartori. Designing a processor from the ground up to allow voltage/reliability tradeoffs. In *HPCA*, 2010.

[5] S. Misailovic, S. Sidiroglou, H. Hoffman, and M. Rinard. Quality of service profiling. In *ICSE*, 2010.

[6] M. K. Qureshi, M. M. Franceschini, L. A. Lastras-Montaño, and J. P. Karidis. Morphable memory system: a robust architecture for exploiting multi-level phase change memories. In *ISCA*, 2010.

[7] H. Yee. A perceptual metric for production testing. In *Journal of Graphics Tools*, 2004.

[8] H. Yee, S. Corley, T. Sauerwein, J. Breidenbach, C. Foster, and J. Tilander. Perceptual image diff, January 2011. `http://pdiff.sourceforge.net`.